DOCUMENT RESUME

ED 323 935 | IR 014 558

AUTHOR	Jonassen, David H.; Wilson, Brent G.
TITLE	Analyzing Automated Instructional Systems: Metaphors from Related Design Professions.
PUB DATE	Feb 90
NOTE	21p.; In: Proceedings of Selected Paper Presentations at the Convention of the Association for Educational Communications and Technology; see IR 014 535.
PUB TYPE	Reports - Research/Technical (143) -- Speeches/Conference Papers (150)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	*Computer Assisted Design; Computer Assisted Manufacturing; *Expert Systems; *Instructional Design; *Microcomputers
IDENTIFIERS	IDD Advisor; ID Expert; IDiom System; Instructional Design Data Base; Park Row ID Tools

ABSTRACT

Noting that automation has had an impact on virtually every manufacturing and information operation in the world, including instructional design (ID), this paper suggests three basic metaphors for automating instructional design activities: (1) computer-aided design and manufacturing (CAD/CAM) systems; (2) expert system advisor systems; and (3) computer-aided systems engineering (CASE) tools used in programming and information systems design. Within this framework, several prototype systems designed to automate different functions of the instructional design process are described: the IDiom system, the Instructional Design Data Base, Park Row ID Tools, ID Expert, and IDD (Instructional Design and Development) Advisor. Prospects for future systems are discussed, including criteria for evaluating automated instructional design systems and the need for continuing research in instructional design problem solving activities. (14 references) (Author/GL)

Title:

Analyzing Automated Instructional Systems: Metaphors from Related Design Professions

Authors:

David H. Jonassen
Brent G. Wilson

Abstract

Automation has impacted virtually every manufacturing and information operation in the world. Instructional design (ID) is no exception. In this article, we suggest three basic metaphors for automating instructional design activities, each typifying different kinds of systems. Within this framework, we describe several prototype systems designed to automate different functions of the ID process. Prospects for future systems are discussed, including criteria for evaluating automated ID systems, and the need for continuing research in ID problem-solving activities.

Analyzing Automated Instructional Design Systems: Metaphors from Related Design Professions

The instructional design (ID) process comprises the procedures, decisions, and knowledge necessary to plan, develop, and evaluate instructional systems and programs. Actually, the process itself can be seen historically as a step toward automation and control of a complex, costly, and diffuse enterprise. It grew out of efforts in the 1950s and 60s within military and university settings to define systematic methods for engineering instructional products, drawing on ideas from general systems theory, psychology, and management science. The thinking was that, at a time when systems design methods were being used to develop sophisticated weapons and manufacturing products, might the same general design principles be used in the design of complex training systems? Models were developed that proceeded systematically from instructional needs and goals toward design of products. Continuing evaluation and testing were core features of these models, as was the idea that all parts of an instructional system should fit together and contribute to the goals of instruction. Today dozens of different ID models are being used to guide the design and development of training products in a variety of public and private sector organizations.

The next major chapter in the history of ID models is presently being written as ID models are being incorporated more completely into integrated computer-based systems for design and production. Presently, several such prototype systems are being developed that further automate ID processes; we will report on some of those systems in this chapter, and provide reference information on other systems. An additional purpose of the chapter is to analyze different kinds of automated systems, to suggest appropriate roles for these systems, and to point to future directions and needed research.

One way of thinking about the problem is to compare the ID process to similar manufacturing or programming design processes. Gibbons and O'Neal (1988a) state:

> Instructional design belongs to the same family of processes that creates building, car, and widget design...During both design processes, the goals of design turn into functions to be performed by the product, which ultimately materialize as

> features. This links features directly to original design goals. Documentation of the design...moves from general expressions of intent and form to detailed expressions of function and finally to exacting specifications for individually catalogued products.

Something can be learned by examining design models in engineering and other design fields. Automated approaches used in these fields can serve as metaphors for ID systems. In the section below, we discuss three different design metaphors that serve as a framework for describing automated ID systems: (1) computer-aided design and manufacturing (CAD/CAM) systems, (2) expert system advisor systems, and (3) computer-aided systems engineering (CASE) tools used in programming and information systems design.

The CAD/CAM Metaphor: ID Environments

CAD/CAM systems include graphics design and data manipulation tools that facilitate the engineering and design of products. The flexibility of these tools allows engineers the freedom to conceptually try out different solutions and quickly and easily make adjustments in a product's structure or dimensions. The engineer using CAD/CAM tools has more power and flexibility than predecessors using traditional tools and instruments.

Not surprisingly, CAD/CAM tools have drastically changed the engineer's job requirements over the last 10 years. Their dramatic rise in use has virtually eliminated the need for traditional draftsmen, drawing tables, squares, and compasses. These changes, of course, have been reflected in schools of engineering around the world, and in the individual engineer's continuing professional education. CAD/CAD tools do not replace the engineer's design expertise; rather they provide a fertile problem-solving environment for professionals to do their jobs better.

The IDioM System

An ID tool designed to emulate a CAD/CAM tool would be designed for professional instructional designers in a way that would allow considerable flexibility and support. As an example, consider the IDioM system developed for Apple Computer, Inc. (Gustafson & Reeves, 1988). IDioM is essentially a shell or superstructure,

written in HyperCard, that oversees ID processes on a Macintosh computer. Within IDioM, instructional design is conceptualized as having 7 main functions, each with several sub-functions or activities. The 7 functions and their associated activities are shown in Table 1.

Insert Table 1 about here.

The IDioM system makes full use of an array of Macintosh applications, such as Excel, More, and PageMaker. At any point, the designer may be within a tool application--say MacProject II, but is also thought to be within a specific IDioM function and activity, such as **select media** within the **Design Function**. The designer has access to three special menus: the general IDioM Menu, the Manage Menu, and the Help Menu. The IDioM Menu provides orienting and guidance information for working within the system and its various functions and activities. The Manage Menu includes further guidance on planning and reporting progress to others. The Help Menu provides information needed for navigating through the system and connecting to the next logical tool or activity. The Help Menu also allows the recording of personal comments, for use in editing and reviewing work.

The IDioM system seems to fit a CAD/CAM type of design model because it is designed primarily to address the needs of professional designers. It imposes a certain structure on development by incorporating basic ID concepts, but still allows considerable flexibility in specifics of design and production. The designer has available a number of data-organizing and production tools to use at any point in the design process. Help and support is provided, but not in a way to severely constrain the designer's personal style. The IDioM system is now being called ID Library by Apple, and is being beta tested at 6 public and private sector locations (Gustafson & Reeves, 1990).

The Instructional Design Data Base

Another prototype that seems to fit the CAD/CAM model is the Instructional Design Data Base developed by designers at WICAT Systems (Gibbons & O'Neal, 1988b). This product is meant to eventually provide ID tools that encompass the entire ID process. Presently a Task Analysis Interface has been developed

for MS-DOS computers (Gibbons, O'Neal, & Monson, 1988) that allows designers to generate either text-based or graphic hierarchical analyses. These analyses are stored within the system and may be sorted, searched, and modified. Although our understanding of the system is based on the operation of the Task Analysis Interface and written descriptions of the larger system, the key to the larger system is the idea of an ID database. Instructional designers work through a successive series of representations of content--needs statements, task analyses, goals and objectives, test items, instructional displays, etc. All of these content representations are interrelated and should be linked together. Changes in one data type should be reflected in changes in the entire system. This core notion of database drives the design and planned capabilities of the overall system.

The CAD/CAM metaphor is apt in describing the Instructional Design Data Base because the system is aimed at the professional designer and offers a measure of flexibility within the constraints of the database. The object of the system is to provide a productive environment within which the designer can perform design-related tasks in an efficient and productive manner. Presently, the ID database concept is being incorporated more completely into another proprietary WICAT product. No information is presently available concerning this new product.

Expert Systems Tools: ID Advisors

Expert systems are decisionmaking tools that make use of reasoning mechanisms developed by artificial intelligence researchers (Grabinger, Wilson, & Jonassen, in press). Most expert systems have a system of IF-THEN rules that constitute the knowledge in a narrow domain. These rules comprise the system's "knowledge base," which is then used to arrive at judgments about what to do in a given situation. The computer program thus becomes an "advisor" to the user, recommending a course of action or choice based on the available data. Expert system advisors typically engage the user in a consultation in which the user responds to questions by providing information about the specific case in hand. The advisor applies that information to its knowledge base and returns with a recommendation or decision. Successful expert systems have been developed for decisionmaking tasks such as diagnosing an illness, choosing a site to drill a well, or troubleshooting a complex piece of equipment (Harmon

and King, 1985).

Another way of thinking about expert systems is as an intelligent job aid. Job aids, in the form of checklists, flowcharts, or step-by-step instructions, are written tools or helps to support job performance; that is also essentially the role of expert system advisors (Welsh and Wilson, 1987). On this view, expert system advisors do not provide a complete working environment for doing the entire job; instead, they are occasionally consulted in addressing specific tasks where the user needs special support or expertise. Job aids may be used by professionals in working with seldom-encountered or complex tasks, but more frequently are designed for the novice worker who needs additional reminding and support for specific tasks. As the novice continues on the job and performs the task repeatedly, the need for the job aid may disappear.

Park Row ID Tools

Park Row is a software development and publishing firm founded by Greg Kearsley, who has developed a series of stand-alone expert system rule bases to perform a variety of instructional design tasks. The Behavioral Objectives system prompts the user to provide information about expected outcomes which it then assembles into an objective. The system then queries the user to analyze the objective and then classifies it using Bloom's taxonomy of objectives. Problem Analysis provides advice on which of several data gathering techniques, such as fault tree analysis, interviews, focus groups, etc., to use in performing a job analysis. CBT Analyst leads the user through a series of decisions to determine whether or not to develop computer-based training. Cost/Benefits Analysis queries the user for information about a prospective training situation. The information is then sent to a spreadsheet to estimate the costs and benefits of developing training. Most of these systems were written in Turbo Pascal and run in an MS-DOS environment with minimal memory requirements.

The Park Row tools fit clearly into the advisor or intelligent job aid category. Each tool provides only analysis and/or advice about the ID process, rather than an actual production environment. The tools may be found useful by professional designers, but would more specifically help the novice designer or technician in need of greater structure and expertise.

CASE TOOLS: Computer-aided ID

Within the last 5-10 years, a category of software tool has attracted attention among systems designers and programmers. CASE (computer-aided systems engineering) tools are programs that automate the programming process (Bigelow, 1988), sometimes referred to as code generators or applications generators. In concept, a user would use a CASE tool by specifying the inputs and desired outputs for a given product, along with other constraints, and the computer would automatically generate fully written code for that product. Although that goal sounds idealistic, some progress has been made in developing systems that compile code to meet specified needs. The idea is similar to a programmer who compiles code blocks from a library of routines rather than starting from scratch on a new project. Much of the programming work has already been done and is available in the form of routines; the job of the programmer is to assemble and fine-tune these routines into a smoothly working program that solves the information/performance problem at hand.

ID Expert

Using the idea of CASE tools for instructional design, we could imagine a product that inputs information about the context, content, learner, etc., and then automatically compiles a specific instructional plan or product. ID Expert, although in its initial stages, is a system that philosophically bears some resemblance to this notion. ID Expert is being developed for the Army Research Institute by M. David Merrill and Zhongmin Li of Utah State University and Human Technology, Inc. of McLean, Virginia under contract with the U.S. Office of Personnel Management, Office of Training Development. It is a combined consultation and production system which provides analysis of content and instructional strategies. In the current phase of development, it is implemented in HyperCrad from Apple and Nexpert from Neuron Data for the Macintosh family of computers. The current version is being tied to the Research Reference Interface, a hypertext system that provides reference information on the theory, policy, empirical support, and logic used by ID Expert. Like ID Expert, RRI is screen-oriented and menu-driven and is written in HyperCard. ID Expert consists of several modules:

- c an information gathering module that queries the user on content to be taught and relevant learner characteristics (such as motivation level and prior knowledge);

- o a reasoning model that uses the information gathered to create and reason with rules in Nexpert's inference engine;

- o a specification module that presents advice on course and content structure and the required instructional strategies;

- o an authoring module for creating instructional sequences; and

- o an explanation module which explains the rule base and connects to RRI.

ID Expert begins by querying the designer about the title, audience attributes, the goal and its attributes, and recommends an appropriate content structure. Next the system queries the designer about specific content attributes and elaborates the content structure while recommending a course organization, instructional strategies, and individual instructional transactions. These transactions can then be created and sequenced in the authoring module.

Unlike most other automated ID systems, ID expert builds in considerable theory. Based on elaboration theory for its course level strategies (Reigeluth & Stein, 1983) and component design theory for developing micro-level strategies (Merrill, 1987), the system offers designers much more specific and concrete support for selecting instructional strategies based on type of learning outcome than is available in other systems. It is in this sense that ID Expert resembles programming CASE tools. The designer supplies information about a given problem, then ID expert is meant to compile the pieces into a coherent instructional strategy. In a sense, ID Expert's ambition is to automate the process to the point that the computer provides essential expertise for designing instructional products, leaving less discretion in the hands of individual designers.

Returning to our discussion of CASE tools for a moment, although the idea of automatic code generators obviously has appeal, forecasts and promises of such tools have exceeded actual delivery performance of

tools developed to date. The problem is complex, particularly because programs must be developed in specific contexts with constraints that are often unique. This means that programs produced automatically following a generic pattern must inevitably be adjusted and changed to fit the new situation. The fine-tuning and changing that must be done often approach in complexity the task of programming from scratch. The ambition of a program like ID Expert may face similar obstacles. The state of research and theory in instructional design may not be able to hold up the platform of decisionmaking upon which ID Expert is built. Although very explicit and guided systems such as ID Expert are a tremendous boon to theory development, their practical use may encounter problems similar to those of programming CASE tools.

Mixing the Metaphors: Hybrid Environments

To this point, system types have been described in idealized terms. Of course, many systems cross over and overlap our metaphors. The IDioM system, for example, contains a connections option off the Help Menu that lends itself well to expert systems technology. The system could track where the designer is within IDioM and anticipate the next needed activity or tool. Thus within an overall production environment for instructional design, expert systems can easily have a place.

The same thing can be said for strongly theory-based products. Different theories might be thought of as templates to be superimposed upon a system: At the designer's request, a different model or template could be selected to drive the design. Thus the rigidity of any particular approach is much less threatening to the professional designer, because it can always be thrown out and replaced by another.

It would seem desirable that a design environment combine the best elements of production tools (graphics, text, and video), database and organization capabilities, and expert systems or hypertext technologies where appropriate. Each of the systems discussed above exhibits different strengths in these areas. In the section below, we turn to our own prototype system, the Instructional Design and Development (IDD) Advisor. Presently the system is designed primarily as an advisor and front end to a database; future plans call for its integration with

production tools to provide a more complete develo ment environment.

IDD Advisor

The IDD Advisor is an on-going project designed to provide intelligent job aid support for assisting an instructional designer, teacher, or course writer to follow the instructional design and development (IDD) process. The IDD Advisor is also an intelligent front end to a database containing the components of the instructional sequence being developed, so that as instructional decisions are made, the results of those decisions are stored in a database.

The IDD Advisor is a structured rule-based expert system with several rule sets. The arrangement of these rule sets is illustrated in the context tree for the IDD Advisor in Figure 1. Each node represents a separate rule set which, in effect, functions as a separate knowledge base. Each rule set draws on fairly generic and well-established ID concepts.

Insert Figure 1 about here.

The first rule system encountered is the executive menu, which queries the user to ascertain if this is a new design project. If the project is new, it executes the project analysis rule system. If the user is adding onto an existing project, the executive menu enables the user to select the rule system that she or he needs to work in.

For new projects, the user would typically work through a project analysis rule set first. This rule set collects information about the project and establishes whether the project is instigated by a performance problem or whether it involves new systems, automatic training, or educational development. If the project is predicated on a performance problem, the project analysis module will access the performance analysis module. The user will provide information about the nature of the performance problem. For instance, a corporate user might identify employee productivity as a problem. If the project is not problem-based, the project analysis module will access the needs assessment rule set. Regardless of the purpose of the project, the project analysis module

saves information about the project.

The performance analysis rule set leads the user through a typical performance analysis: identify the goal performance state, the current state, determines the discrepancy, determines the cause and classifies the cause of the performance problem. If the problem is caused by motivational or environmental problems, the user would exit the system. The performance analysis rule base would help the user identify the cause and whether or not training is the proper solution. In the future, motivational and environmental problem rule sets may be developed. If the problem is determined to be a skill/knowledge problem, then the problems are recorded and indexed in the database, and the user is led to the task analysis module.

The needs assessment rule set is accessed if the project analysis determines that the project is not problem-based. If that is the case, users need to decide which data collection instrument is appropriate for conducting the needs assessment. Should they use a questionnaire, interview, focused group or brainstorming session? Users are led through the needs assessment technique which helps them decide why type of instrument is appropriate for their need. Having determined the instrument type, they need to select the question form and question type in order to construct the questions. The outcome of these modules is a needs assessment instrument with the questions identified. This information is then stored in the database. After collecting the data, users would access the needs assessment interpretation rule set to help them interpret the outcome. The users would then be led to the task analysis module.

The task analysis process has three rule sets that assist the user. The first module helps the user select the most appropriate task analysis technique from a list of 27 found in Jonassen, Hannum & Tessmer (1989). A sample of the rules in this rule set is illustrated in Figure 2. The technique rule set may access the tool rule set to determine the most appropriate information gathering tools. Finally, the task analysis rule set will help users to interpret and analyze the information they collect. The sub-tasks will be stored in the database.

Insert Figure 2 about here.

Following the task analysis, the objective writing and test item writing rule sets will access the tasks stored in the database and help the users to convert them to objectives, classify those objectives, and write test items that measure those objectives.

Having developed the test items, designers would then access the instructional design modules, which would help them to select the necessary presentation forms and instructional strategies for the sequence of tasks/objectives determined in the previous modules. Again, the rule sets will interact with the database by pulling the information out of the database and using it to help the users to select the instructional lesson components.

Finally, designers could access the delivery module to access advice on selecting the appropriate delivery system for presenting the instructional components. The system may be expanded to include a message design module in the future.

The prototype knowledge bases run under VP Expert from Paperback Software. VP Expert is a structured, rule-based system with considerable power and flexibility. The information base is being created in dBase III+ from Ashton-Tate. VP Expert can interface with dBase files in order to function as an intelligent front end to the database. Both software packages run on MS-DOS personal computers (IBM or compatible). The reasons for selecting the software packages currently being used to develop IDD Advisor are pragmatic. They are inexpensive, readily available, and commonly used. That makes the system more useful to a wider range of users.

Harmon (1986) describes a seven step process for design, developing, and implementing expert systems.

1) Front end analysis - performance problem and cost benefit

2) Task analysis

3) Prototype development - development of a small version to test feasibility and desirability of system

4) System development - develop entire system

5) Field test the system

6) Implementation - disseminate the system

7) Maintenance - upgrade the system and provide user support

The IDD Advisor is currently at the fourth step in this process --- system development. The task analysis yielded the context tree shown in Figure 1. A prototype of two of the rule sets was demonstrated (Jonassen, 1988). The remainder of the rule sets and the database files need to be developed.

Needed Research on Instructional Design

All of the systems described above incorporate ID concepts and procedures that have proven their worth over the past 25 years in field-based development projects. At the same time, automating these procedures affords a new opportunity to validate the effects of these procedures. With the computer to assist in data gathering and storage, differences between systems can be explored. This in turn can refine our knowledge base regarding effective ID procedures.

While generic ID models have been somewhat validated by practical experience, there is a surprising lack of empirically derived knowledge about the problem-solving heuristics of professional instructional designers. How do individual designers do their work within the overall ID framework? How closely do they follow prescriptions from the models; when do they use their intuition and professional judgment to override conventional ID recommendations? What kinds of learning, management, and design principles do they actually incorporate at various points in the design process? Virtually no formal research has been completed to study these questions, yet an understanding of designers' problem-solving strategies seems extremely relevant to the design of automated ID systems. We also need to be concerned about designers' use of tools: How do designers use computers and other reference/production/organization tools in their work? The designer/computer interface needs to be more thoroughly studied before automated ID systems can be optimized.

Rather than be stymied by our lack of knowledge in these areas, we should see the advent of automated ID systems as opportunities to conduct in-depth studies

and examine processes in closer detail. Thus, in the long run, automated ID systems will not only incorporate our present knowledge of ID, but also contribute to that knowledge in the future. In the same way that cognitive science researchers develop computer-based models of cognitive processes, instructional designers will come to view the development of automated ID systems as a valuable method of research and a way to contribute to our knowledge base about instruction.

References

Bigelow, J. (1988, March). Hypertext and CASE. IEEE Software, 23 - 27.

Gibbons, A. S., & O'Neal, A. F. (1988, November). New work surfaces for instructional system design. Paper presented at the annual meeting of the Association for the Development of Computer-based Instructional Systems (ADCIS), Philadelphia.

Gibbons, A. S., & O'Neal, A. F. (1988, November). The concept and operation of an instructional design data base. Orem, UT: WICAT Systems, Inc., unpublished manuscript.

Gibbons, A., O'Neal, F., & Monson, L. (1988, March). Introduction to the Task Analysis Interface. Orem, UT: WICAT Systems, Inc., unpublished manuscript.

Grabinger, R. S., Wilson, B. G., & Jonassen, D. H. (in press). Designing expert systems for education and training. New York: Praeger Press.

Gustafson, K. L., & Reeves, T. C. (1988, November). IDioM: An instructional design workstation for business, sales, and technical training. Paper presented at the "Computer and Video in Company Training" conference co-sponsored by the Gottlieb Duttweiler Institute, New York University, and the Utrecht Academy of Arts, Zurich, Switzerland.

Gustafson, K. L., & Reeves, T. C. (1990, January). IDioM: An instructional design environment. Paper presented at the meeting of the Association for Educational Communications and Technology, Anaheim, CA.

Harmon, P. (1986). Developing a small expert system. Expert System Strategies, 2(1), 1-6.

Harmon, P. (1987). Intelligent job aids: How AI will change training in the next five years. In G. Kearsley (Ed.), Artificial intelligence & instruction. Reading, MA: Addison-Wesley.

Harmon, P. & King, D. (1985). Expert systems: Artificial intelligence in business. New York :Wiley.

Jonassen, D.H., Hannum, W.H., & Tessmer, M. (in press). Handbook of task analysis procedures. New York: Praeger.

Merrill, M. D. (1989). Component design theory. Instructional Science.

Reigeluth, C. M., & Stein, R. (1983). Elaboration theory. In C. M. Reigeluth (ed.), Instructional-design theories and models: An overview of their current status. Hillsdale, NJ: Erlbaum.

Welsh, J., & Wilson, B. G. (1987). Intelligent job aids. Journal of Instructional Development.

Table and Figure Captions

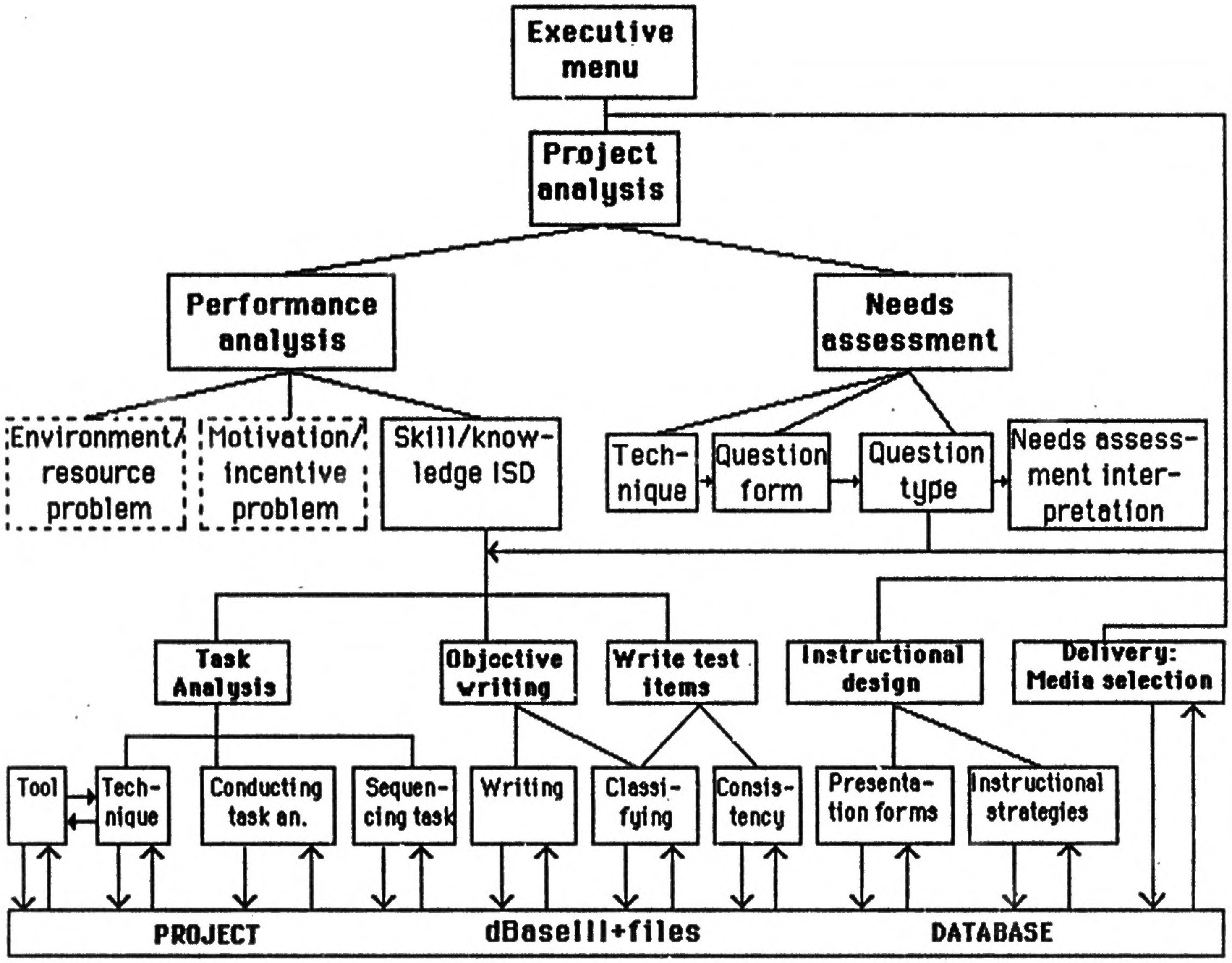
Table 1. IDioM Functions and Activities (from Gustafson & Reeves, 1988).

Figure 2. Context tree for the IDD Advisor. Shaded boxes are presently not developed.

Figure 3. Sample rules from the IDD Advisor.

FUNCTION	ACTIVITY
ANALYZE	Conduct needs and goal assessment Define learner characteristics Conduct task analysis Examine existing materials Conduct financial analysis
DESIGN	Specify objectives Sequence objectives Determine strategy Select media Prepare assessment
DEVELOP	Define lesson treatment Outline content Prepare training materials Integrate interactive media
EVALUATE	Collect project documentation data Assess worth of training objectives Conduct formative evaluation Assess immediate effectiveness Evaluate training impact Estimate cost effectiveness
IMPLEMENT	Prepare implementation plan Communicate with Training Center Support implementation Train the trainer
PRODUCE	Prepare component list and estimate Prepare packaging Specify and order components Assemble package Debrief
MANAGE	Estimate project scope Start up with project

Table 1. Functions and Activities from the IDioM environment, presently called the ID Library (from Gustafson & Reeves, 1988).


Executive menu
Project analysis
Performance analysis
Needs assessment
Environment/ resource problem
Motivation/ incentive problem
Skill/know-ledge ISD
Tech-nique
Question form
Question type
Needs assess-ment inter-pretation
Task Analysis
Objective writing
Write test items
Instructional design
Delivery: Media selection
Tool
Tech-nique
Conducting task an.
Sequen-cing task
Writing
Classi-fying
Consis-tency
Presenta-tion forms
Instructional strategies
PROJECT
dBaseIII+files
DATABASE

327

Figure 2
Rules for Selecting Task Analysis Techniques

IF	Type-Job AND
	Scope-Single AND
	Function-Sequence AND
	Cost_Time-Low AND
	Expertise-High
THEN	Technique-Learning_Contingency_Analysis;

RULE 12

IF	Type-Job AND
	Scope-Single AND
	Function-Description AND
	Cost_Time-Low AND
	Expertise-Low
THEN	Technique-Behavioral_Analysis;

RULE 13

IF	Type-Job AND
	Scope-Single AND
	Function-Description AND
	Cost_Time-High AND
	Expertise-High
THEN	Technique-Functional_Job_Analysis;

RULE14

IF	Type-Job AND
	Scope-Single AND
	Function-Description AND
	Cost_Time-Low AND
	Expertise-High
THEN	Technique-Learning_Contingency_Analysis;

RULE 15

IF	Type-Job AND
	Scope-Single AND
	Function-Description AND
	Cost_Time-Low AND
	Expertise-Low
THEN	Technique-Methods_Analysis;

RULE 16

IF	Type-Job AND
	Scope-Single AND
	Function-Description AND
	Cost_Time-Low AND
	Expertise-Low
THEN	Technique-Task Description;